ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

This document describes enhancements to the ATM virtual circuit (VC) bundle management feature, which allows you to configure multiple VCs that have different quality of service (QoS) characteristics between any pair of ATM-connected routers that support this feature. VC bundle management allows multiple VCs with various QoS settings to be directed to the same destination and to map traffic to the VCs based on protocol criteria associated with the traffic. Three experimental (EXP) bits in the Multiprotocol Label Switching (MPLS) packets determine which VC in the bundle to use to forward the packets.

Feature History for the ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection Feature

Release	Modification
12.2(8)T	This feature was introduced.
12.0(23)S	This feature was made available on the 8-port OC-3 STM-1 ATM line card for Cisco 12000 series Internet routers.
12.0(29)S	This feature was integrated into Cisco IOS Release 12.0(29)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

This feature requires ATM VC management, Cisco Express Forwarding (CEF), and Forwarding Information Base (FIB) and Tag Forwarding Information Base (TFIB) switching functionality.

Restrictions for ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

- The router at the remote end must be running a version of Cisco IOS software that supports MPLS and ATM VC management.
- This feature is not supported on either the ATM interface processor (AIP) or the ATM Lite port adapter (PA-A1).

Information About ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

You need to understand the concepts in the following sections to configure the MPLS EXP Bits Based ATM PVC Bundles VC Selection feature:

ATM VC Bundle Management

The MPLS EXP Bits Based ATM PVC Bundles VC Selection feature is an extension to the IP to ATM Class of Service feature suite. The IP to ATM Class of Service feature suite, using VC support and bundle management, maps QoS characteristics between IP and ATM. It provides customers that have multiple VCs (with varying qualities of service to the same destination) the ability to build a QoS differentiated network.

The IP to ATM Class of Service feature suite allows customers to use IP precedence level as the selection criterion for packet forwarding. This feature provides customers with the option of using the MPLS experimental level as an additional selection criterion for packet forwarding.

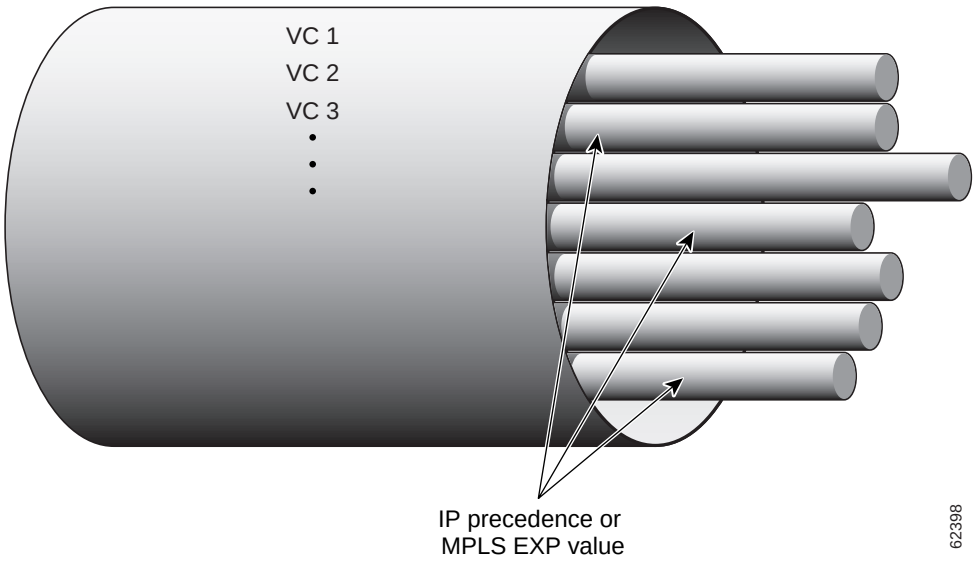
Note If a selection criterion for packet forwarding is not selected (that is, if the packet is unlabeled), this feature uses the IP precedence level as the default selection criterion.

For more information about the IP to ATM Class of Service feature suite, see the "Related Documents" section on page 24.

ATM VC Bundle Configuration

ATM VC bundle management allows you to configure multiple VCs that have different QoS characteristics between any pair of ATM-connected routers. As shown in Figure 1, these VCs are grouped in a bundle and are referred to as *bundle members*.

Figure 1 ATM VC Bundle



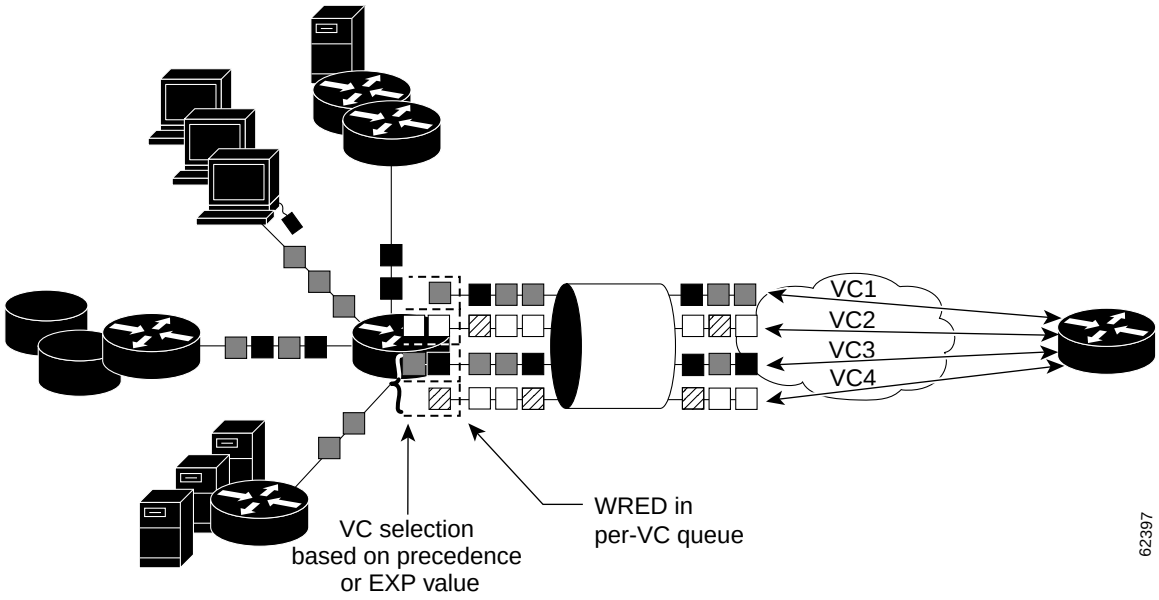
ATM VC bundle management allows you to define an ATM VC bundle and add VCs to it. Each VC of a bundle has its own ATM traffic class and ATM traffic parameters. You can apply attributes and characteristics to discrete VC bundle members, or you can apply them collectively at the bundle level.

Using VC bundles, you can create differentiated service by flexibly distributing MPLS EXP levels over the different VC bundle members. You can map a single MPLS EXP level, or a range of these levels, to each discrete VC in the bundle, thereby enabling individual VCs in the bundle to carry packets marked with different MPLS EXP levels. You can use Weighted Random Early Detection (WRED) or distributed WRED (dWRED) to further differentiate service across traffic that has different MPLS EXP levels.

To determine which VC in the bundle to use to forward a packet to its destination, the ATM VC bundle management software matches MPLS EXP levels between packets and VCs (see Figure 2). IP traffic is sent to the next hop address for the bundle because all VCs in a bundle share the same destination, but the VC used to carry a packet depends on the value set for that packet in the MPLS EXP level of the type of service (ToS) byte of its header. The ATM VC bundle management software matches the MPLS EXP level of the packet to the MPLS EXP levels assigned to a VC, sending the packet out on the appropriate VC.

Moreover, the ATM VC bundle management software allows you to configure how traffic will be redirected when the VC to which the packet was initially directed goes down. Figure 2 illustrates how the ATM VC bundle management software determines which permanent virtual circuit (PVC) bundle member to use to carry a packet and how WRED (or dWRED) is used to differentiate traffic on the same VC.

Figure 2 ATM VC Bundle PVC Selection for Packet Transfer



The support of multiple parallel ATM VCs allows you to create stronger service differentiation at the IP layer. For example, you might want to configure the network to provide IP traffic belonging to real-time class of service (CoS) such as Voice over IP traffic on an ATM VC with strict constraints on constant bit rate (CBR) or variable bit rate real-time (VBR-rt), while also allowing the network to transport nonreal-time traffic over a more elastic ATM unspecified bit rate (UBR) PVC. UBR is effectively the ATM version of best-effort service. Using a multiple parallel ATM VC configuration allows you to make full use of your network capacity.

Benefits of ATM VC Bundle Management

ATM VC bundle management was designed to provide a true working solution to class-based services, without the investment of new ATM network infrastructures. It allows networks to offer different service classes (sometimes termed differential service classes) across the entire WAN, not just the routed portion. Mission-critical applications can be given exceptional service during periods of high network usage and congestion. In addition, noncritical traffic can be restricted in its network usage, ensuring greater QoS for more important traffic and user types.

ATM VC bundle management gives customers the option of using the MPLS EXP level, in addition to IP precedence, as a selection criterion for packet forwarding.

VC Bundle Management Supported Features

The following features are supported on an MPLS over VC bundle:

- PVC support only (no switched virtual circuits or SVCs):
  - Support for multipoint and point-to-point subinterfaces.
  - Support for AAL5SNAP (RFC1483 bridging) and multiplex (MUX) type VCs encapsulation.
  - Use of static mapping and Inverse Address Resolution Protocol (Inverse ARP) for the next hop protocol address (supported on multipoint subinterfaces only).
  - PVCs associated with VC bundles through explicit configuration.
  - Use of Interim Local Management Interface (ILMI) and Operation, Administration, and Maintenance (OAM) functionality in the PVC management feature for PVC failure detection.
- VC selection within the bundle:
  - Uses three EXP bits in the MPLS header to define the precedence levels, with level 7 being the highest for MPLS traffic.
  - No automapping of VCs to precedence levels can be done. The user must use the **mpls experimental** command under each member VC to explicitly specify the mapping.
  - Multiple precedence levels can be mapped to one VC.
  - Packets with the PAK_PRIORITY_CRUCIAL flag set go on a high precedence (level 6) VC. These packets include IP routing packets such as Intermediate System-Intermediate System (IS-IS) packets for integrated IP routing. Label Distribution Protocol (LDP) and Tag Distribution Protocol (TDP) messages, and Inverse ARP packets also use a precedence level 6 VC. However, OAM cells still flow in the individual VC to detect PVC failures, although the PAK_PRIORITY_CRUCIAL flag is set for them.
  - Regular **ping** commands use the lowest precedence (level 0) VC. If other protocols such as Internetwork Packet Exchange (IPX) are configured in the bundle, they will also use the lowest precedence level VC for their traffic.
- ATM Inverse ARP:
  - Inverse ARP is viewed as a parameter at the bundle level and can be enabled or disabled only for the bundle, not for individual VCs in the bundle.
  - The PAK_PRIORITY_CRUCIAL flag is set in each ATM Inverse ARP packet and the packets use the precedence level 6 VC.
  - Inverse ARP for other protocols such as IPX is off by default unless it is configured in the bundle.
- Broadcast and multicast:
  - Broadcasting can be turned on or off at the bundle level, not at the individual VC level in the bundle.
  - Pseudobroadcasting is used for forwarding the broadcast traffic.
  - VC selection for the broadcast traffic is based on the precedence levels of the broadcast packets.
- Bundle management:
  - According to the protected group rule, when all members in the protected group fail, the bundle is declared down.
  - According to the protected VC rule, when a protected VC goes down, the bundle goes down.
  - A VC can be a standalone VC or belong to only one bundle.

  - When a bundle goes down, no traffic should be forwarded out the bundle, even if some of the VCs in the bundle are still up.
  - In VC bumping, each bundle member can specify if bumping is allowed. If bumping is allowed, the next lower precedence level VC is selected when a VC goes down. This is the implicit bumping rule. Traffic is restored to the original VC when it comes back.
  - In explicit VC bumping, a VC can specify to which precedence level it wants to bump its traffic when it goes down. Only one precedence level can be specified for bumping. If the VC that carries the bumped traffic also fails, the traffic will follow the bumping rules specified for that VC.
  - In reject bumping, a VC may also be configured not to accept the bumped traffic.
  - When no alternate VC can be found for some bumped traffic, the bundle has to be declared down.
  - To avoid bringing down a bundle because of a failure of the lowest precedence VC, configure explicit bumping on the lowest precedence VC.
  - Bundle status attributes and their current status for each VC in the bundle can be displayed in a tabular form using EXEC commands.
  - Bundle statistics are the same statistics provided for VC that have been aggregated for a VC bundle.
  - Bundle debugging commands, when enabled, print bundle events and bundle errors.
- Packet forwarding:
  - There are four possible paths for MPLS packet forwarding over the VC bundle: IP to MPLS, MPLS to MPLS, MPLS to IP, and locally generated packets.
  - Process switching is used for locally generated packets.
  - CEF FIB switching is used for the IP to MPLS path.
  - CEF TFIB switching is used for the MPLS to MPLS and MPLS to IP paths.
  - No fast switching is supported for transit IP packets. The fast switching path does not classify IP packets based on their precedence levels.
  - VC bundle configuration is already added to handle the IP VC bundle feature and may be used without any modification.

How to Configure ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

This section contains the following procedures:

- Configuring MPLS and Creating a VC Bundle, page 7 (required)
- Configuring the Bundle-Level Protocol, page 9 (required)
- Configuring Parameters on a VC Bundle Member Directly, page 9 (optional)
- Configuring a VC Class and Applying Parameters to a Bundle, page 11 (optional)
- Attaching a Class to a Bundle, page 14 (optional)
- Configuring a VC Bundle at the Subinterface Level, page 15 (optional)
- Assigning VC and Bundle Attributes, page 17 (optional)

- Verifying ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection Configuration, page 20 (optional)

Configuring MPLS and Creating a VC Bundle

Perform the following steps to enable MPLS and create a VC bundle. When you create a VC bundle, you enter bundle configuration mode, in which you can assign attributes and parameters to the bundle and to all of its member VCs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **mpls ldp advertise-labels**
5. **interface atm** *interface-number*[.*subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
6. **ip address** *ip-address mask*
7. **mpls ip**
8. **bundle** *bundle-name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`ip cef [distributed]`
	
	Example:
	`Router(config)# ip cef`
		Enables CEF, which is needed for MPLS.
		Note This command without the optional keyword enables CEF on the Route Processor (RP) card. The optional **distributed** keyword is used to enable distributed CEF (dCEF) for the Versatile Interface Processor (VIP)-based platforms.

Step 4	`mpls ldp advertise-labels`
	
	Example:
	`Router(config)# mpls ldp advertise-labels`
		Controls the distribution of locally assigned, incoming labels by means of the LDP, and allows TDP neighbors to exchange messages between them.

	Command or Action	Purpose
Step 5	**interface atm** *interface-number*[.*subinterface-number* {**mpls** | **multipoint** | **point-to-point**}] **Example:** `Router(config)# interface atm 0/0/3`	Configures an ATM interface and enters interface configuration mode.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.13.11.3 255.255.0.0`	Sets the IP address for an interface.
Step 7	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables MPLS forwarding of IP packets along normally routed paths for the platform.
Step 8	**bundle** *bundle-name* **Example:** `Router(config-if)# bundle new-york`	Creates or modifies a bundle and enters bundle configuration mode. The prompt changes to the following: `Router(config-if-atm-bundle)#`

What to Do Next

Decide whether you want to configure the VC bundle member directly or use a VC class attached to a bundle.

You can apply parameters (or attributes) to bundles either by applying the parameters directly to the bundle or by applying the parameters to a VC class assigned to the bundle.

Applying parameters by using VC classes assigned to the bundle allows you to apply multiple parameters at once because you apply the VC class to the bundle and to all of its VC members. This method allows you to apply a parameter across all VCs for the bundle, after which (for some parameters) you can later modify that parameter for individual VCs. After configuring the parameters for the VC class, you need to attach the VC class to the bundle.

To configure the VC bundle member directly, complete the procedure in the "Configuring Parameters on a VC Bundle Member Directly" section on page 9. To use a VC class attached to a bundle, instead complete the procedures in both the "Configuring a VC Class and Applying Parameters to a Bundle" section and the "Attaching a Class to a Bundle" section on page 14.

Parameters applied directly to a bundle take priority over those applied to VC classes assigned to the bundle, and the steps for this task are in the "Configuring the Bundle-Level Protocol" section on page 9. Parameters applied to VC classes assigned to the bundle take priority over those applied to individual VCs.

Note Note that some parameters applied through a VC class or directly to the bundle can be superseded by commands that you apply directly to individual VCs in bundle-VC configuration mode.

Configuring the Bundle-Level Protocol

Perform the following steps to configure a protocol that applies to the bundle and to all of its members. The commands in these steps are entered in bundle configuration mode.

SUMMARY STEPS

1. **protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**]
2. **encapsulation** [**aal5mux** | **aal5snap**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**] **Example:** `Router(config-if-atm-bundle)# protocol clns 49.0000.0000.0000.3333.00 broadcast`	Configures a static map (the map statement for the bundle) for an ATM PVC, SVC, or VC class. • *protocol*—Networking protocol. • *protocol-address*—Destination address that is being mapped to a PVC. • **inarp**—(Valid only for IP and IPX protocols on PVCs) Enables Inverse ARP on an ATM PVC. If you specify a protocol address instead of the **inarp** keyword, Inverse ARP is automatically disabled for that protocol. • [**no**] **broadcast**—Indicates that this map entry is used when the corresponding protocol sends broadcast packets to the interface. **Note** Pseudobroadcasting is supported. The **broadcast** keyword of the **protocol** command takes precedence if you previously configured the **broadcast** command on the ATM PVC or SVC.
Step 2	**encapsulation** [**aal5mux** | **aal5snap**] **Example:** `Router(config-if-atm-bundle)# encapsulation aal5snap`	Configures the ATM adaptation layer (AAL) and encapsulation type for every VC in the bundle. • **aal5mux**—AAL and encapsulation type for multiplex (MUX) type VCs. A protocol must be specified when using this encapsulation type. • **aal5snap**—AAL and encapsulation type that supports Inverse ARP.

Configuring Parameters on a VC Bundle Member Directly

Perform the following steps to configure parameters on an individual VC bundle member directly. The commands in these steps are entered in bundle configuration mode.

SUMMARY STEPS

1. **ubr** *pcr*
2. **ubr+** *pcr mcr*

3. **vbr-nrt** *pcr scr* [*mbs*]
4. **mpls experimental** [**other** | *range*]
5. **bump** {**implicit** | **explicit** *precedence-level* | **traffic**}
6. **protect** {**group** | **vc**}
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ubr** *pcr* **Example:** `Router(config-if-atm-bundle)# ubr 10000`	Configures UBR QoS and specifies the output peak cell rate (PCR) for the VC bundle member.
Step 2	**ubr+** *pcr mcr* **Example:** `Router(config-if-atm-bundle)# ubr+ 10000 3000`	Configures UBR+ QoS with a PCR and a minimum cell rate (MCR) for the VC bundle member.
Step 3	**vbr-nrt** *pcr scr* [*mbs*] **Example:** `Router(config-if-atm-bundle)# vbr-nrt 20000 10000 32`	Configures variable bit rate non-real-time (VBR-nrt) QoS with a PCR, a sustaining cell rate (SCR), and maximum burst size (MBS).
Step 4	**mpls experimental** [**other** | *range*] **Example:** `Router(config-if-atm-bundle)# mpls experimental 7`	Configures MPLS EXP levels for a VC class that can be assigned to a VC bundle and thus applied to all VC members of that bundle. • This command is ignored if the class that contains it is not attached to a bundle member. • **other**—Any precedence level from 0 to 7 not explicitly configured. • *range*—An MPLS EXP level specified as a number or a range of numbers. Ranges can be expressed with a hyphen (2-5, for example), and numbers and ranges can be expressed in groups separated by commas; 1, 3, 5-7, for example.

	Command or Action	Purpose
Step 5	**bump** {**implicit** | **explicit** *precedence-level* | **traffic**} **Example:** `Router(config-if-atm-bundle)# bump explicit 7`	Configures the bumping rules and applies only to bundle members. • **implicit**—Applies the implicit bumping rule, which is the default, to a single VC or PVC bundle member or to all VCs in the bundle (VC-class mode). The implicit bumping rule stipulates that bumped traffic is to be carried by a VC or PVC with a lower precedence level. • **explicit** *precedence-level*—Specifies a precedence level from 0 to 7 for the traffic to be bumped to. • **traffic**—Specifies that the VC or PVC accepts bumped traffic (the default condition). The **no** form of this command stipulates that the VC or PVC does not accept any bumped traffic.
Step 6	**protect** {**group** | **vc**} **Example:** `Router(config-if-atm-bundle)# protect vc`	Configures a VC class with protected group or protected VC status for application to a VC bundle member. • This command makes a bundle member part of the protected group of a bundle or a protected VC in a bundle. • **group**—Configures the VC or PVC bundle member as part of the protected group of the bundle. • **vc**—Configures the VC or PVC member as individually protected.
Step 7	**exit** **Example:** `Router(config-if-atm-bundle)# exit`	Exits the current configuration mode. • Continue entering **exit** at the prompt to exit each configuration mode.

Configuring a VC Class and Applying Parameters to a Bundle

This section describes the task to configure a VC class to contain commands that configure all VC members of a bundle when the class is applied to that bundle. The parameters are applied in VC-class configuration mode. Use the **vc-class atm** command in global configuration mode to enter the VC-class configuration mode.

Commands Ignored in a VC Class Bundle

When a VC is part of a bundle, some of the VC configuration in the VC class will no longer be applicable to the VC and will be ignored. The inheritance rule for VCs in VC bundles follows this order: VC configuration, bundle configuration, subinterface configuration. In VC mode and bundle mode, the configuration with the individual command takes precedence over the configuration with the **class** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **oam-bundle** [**manage**] [*frequency*]
5. **mpls experimental** [**other** | *range*]
6. **bump** {**implicit** | **explicit** *precedence-level* | **traffic**}
7. **protect** {**group** | **vc**}
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Router(config)# vc-class atm bundle-class`	Creates a VC class for an ATM interface and enters VC-class configuration mode.
Step 4	**oam-bundle** [**manage**] [*frequency*] **Example:** `Router(config-vc-class)# oam-bundle manage 3`	Enables end-to-end F5 OAM loopback cell generation and determines whether the bundle is OAM managed, that is, whether every VC in the bundle is OAM managed. There is no effect if the VC class that contains this command is not attached to a bundle. • **manage**—Enables OAM management. If this keyword is omitted, loopback cells are sent, but the bundle is not managed. • *frequency*—Seconds between transmitted OAM loopback cells. Default is 10 seconds.

	Command or Action	Purpose
Step 5	**mpls experimental** [**other** | *range*] **Example:** `Router(config-vc-class)# mpls experimental 7`	Configures MPLS EXP levels for a VC class that can be assigned to a VC bundle and thus applied to all VC members of that bundle. • This command is ignored if the class that contains it is not attached to a bundle member. • **other**—Any precedence level from 0 to 7 not explicitly configured. • *range*—An MPLS EXP level specified as a number or a range of numbers. Ranges can be expressed with a hyphen (2-5, for example), and numbers and ranges can be expressed in groups separated by commas; 1, 3, 5-7, for example.
Step 6	**bump** {**implicit** | **explicit** *precedence-level* | **traffic**} **Example:** `Router(config-vc-class)# no bump traffic`	Configures the bumping rules and applies only to bundle members. • **implicit**—Applies the implicit bumping rule, which is the default, to a single VC or PVC bundle member or to all VCs in the bundle (VC-class mode). The implicit bumping rule stipulates that bumped traffic is to be carried by a VC or PVC with a lower precedence level. • **explicit** *precedence-level*—Specifies a precedence level from 0 to 7 for the traffic to be bumped to. • **traffic**—Specifies that the VC or PVC accepts bumped traffic (the default condition). The **no** form of this command stipulates that the VC or PVC does not accept any bumped traffic.
Step 7	**protect** {**group** | **vc**} **Example:** `Router(config-vc-class)# protect vc`	Configures a VC class with protected group or protected VC status for application to a VC bundle member. • This command makes a bundle member part of the protected group of a bundle or a protected VC in a bundle. • **group**—Configures the VC or PVC bundle member as part of the protected group of the bundle. • **vc**—Configures the VC or PVC member as individually protected.
Step 8	**exit** **Example:** `Router(config-vc-class)# exit`	Exits the current configuration mode. • Continue entering **exit** at the prompt to exit each configuration mode.

Attaching a Class to a Bundle

Perform the following steps to attach a VC class containing bundle-level configuration commands to a bundle. Enter the **bundle** command in global configuration mode to enter bundle configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bundle** *bundle-name*
4. **class-bundle** *vc-class-name*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**bundle** *bundle-name* **Example:** `Router(config)# bundle new-york`	Creates or modifies a bundle and enters bundle configuration mode. • *bundle-name*—Specifies the name of the bundle to be created. Name is limited is 16 characters.
Step 4	**class-bundle** *vc-class-name* **Example:** `Router(config-if-atm-bundle)# class-bundle class1`	Configures a bundle with the bundle-level commands contained in the specified VC class.
Step 5	**exit** **Example:** `Router(config-if-atm-bundle)# exit`	Exits the current configuration mode. • Continue entering **exit** at the prompt to exit each configuration mode.

Configuring a VC Bundle at the Subinterface Level

The commands in the following steps can be used to configure a bundle at the subinterface configuration level. The bundle submode is activated by entering the **bundle** command. This mode is similar to the VC mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-name*
4. **bundle** *bundle-name*
5. **encapsulation** [**aal5mux** | **aal5snap**]
6. **protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**]
7. **class** *class-name*
8. **ubr** *pcr*
9. **ubr+** *pcr mcr*
10. **vbr-nrt** *pcr scr* [*mbs*]
11. **oam-bundle** [**manage**] [*frequency*]
12. **oam retry** [*up-count*] [*down-count*] [*retry-frequency*]
13. **inarp** [*minutes*]
14. **broadcast**
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *interface-number* **Example:** `Router(config-if)# interface atm 0/0/3`	Configures an ATM interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**bundle** *bundle-name* **Example:** `Router(config-if)# bundle new-york`	Creates or modifies a bundle and enters bundle configuration mode. • *bundle-name*—Specifies the name of the bundle to be created. Name is limited is 16 characters.
Step 5	**encapsulation** [**aal5mux** | **aal5snap**] **Example:** `Router(config-atm-bundle)# encapsulation aal5snap`	Configures the AAL and encapsulation type for every VC in the bundle. • **aal5mux**—AAL and encapsulation type for multiplex (MUX) type VCs. A protocol must be specified when using this encapsulation type. • **aal5snap**—AAL and encapsulation type that supports Inverse ARP.
Step 6	**protocol** *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**] **Example:** `Router(config-atm-bundle)# protocol clns 49.0000.0000.0000.3333.00 broadcast`	Configures a static map (the map statement for the bundle) for an ATM PVC, SVC, or VC class. • *protocol*—Networking protocol. • *protocol-address*—Destination address that is being mapped to a PVC. • **inarp**—(Valid only for IP and IPX protocols on PVCs) Enables Inverse ARP on an ATM PVC. If you specify a protocol address instead of the **inarp** keyword, Inverse ARP is automatically disabled for that protocol. • [**no**] **broadcast**—Indicates that this map entry is used when the corresponding protocol sends broadcast packets to the interface. **Note** Pseudobroadcasting is supported. The **broadcast** keyword of the **protocol** command takes precedence if you previously configured the **broadcast** command on the ATM PVC or SVC.
Step 7	**class** *class-name* **Example:** `Router(config-atm-bundle)# class control-class`	Attaches a named VC class to this bundle.
Step 8	**ubr** *pcr* **Example:** `Router(config-atm-bundle)# ubr 10000`	Configures UBR QoS and specifies the output PCR for the VC bundle member.
Step 9	**ubr+** *pcr mcr* **Example:** `Router(config-atm-bundle)# ubr+ 10000 3000`	Configures UBR+ QoS with a PCR and a MCR for the VC bundle member.
Step 10	**vbr-nrt** *pcr scr* [*mbs*] **Example:** `Router(config-atm-bundle)# vbr-nrt 20000 10000 32`	Configures VBR-nrt QoS with a PCR, an SCR, and MBS.

	Command or Action	Purpose
Step 11	**oam-bundle** [**manage**] [*frequency*] **Example:** `Router(config-atm-bundle)# oam-bundle manage 6`	Enables OAM for every VC in the bundle. • **manage**—Enables OAM management. If this keyword is omitted, loopback cells are sent, but the bundle is not managed. • *frequency*—Seconds between transmitted OAM loopback cells. Default is 10 seconds.
Step 12	**oam retry** [*up-count*] [*down-count*] [*retry-frequency*] **Example:** `Router(config-atm-bundle)# oam retry 5 3 10`	Configures OAM parameters for every VC in the bundle. • *up-count*—Consecutive end-to-end F5 OAM loopback cell responses that must be received to change a connection state to up. Default is 3. • *down-count*—Consecutive end-to-end F5 OAM loopback cell responses that are not received to change a PVC state to down. Default is 5. • *retry-frequency*—Frequency (in seconds) that end-to-end F5 OAM loopback cells are transmitted when a change in the up/down state is being verified. Default is 1 second.
Step 13	**inarp** [*minutes*] **Example:** `Router(config-atm-bundle)# inarp 1`	Configures the Inverse ARP time period. • Default is 15 minutes.
Step 14	**broadcast** **Example:** `Router(config-atm-bundle)# broadcast`	Enables broadcast forwarding on this bundle.
Step 15	**exit** **Example:** `Router(config-atm-bundle)# exit`	Exits the current configuration mode. • Continue entering **exit** at the prompt to exit each configuration mode.

Assigning VC and Bundle Attributes

The **pvc-bundle** command activates the bundle-VC configuration mode, in which specific VC and bundle member attributes can be assigned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pvc-bundle** *pvc-name* [**vpi**/][**vci**]
4. **class** *class-name*
5. **ubr** *pcr*
6. **ubr+** *pcr mcr*

7. **vbr-nrt** *pcr scr* [*mbs*]
8. **mpls experimental** [**other** | *range*]
9. **bump** {**implicit** | **explicit** *precedence-level* | *traffic*}
10. **protect** {**group** | **vc**}
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pvc-bundle** *pvc-name* [**vpi**/][**vci**] **Example:** `Router(config)# pvc-bundle ny-control 207`	Adds a VC to a bundle as a member and enters bundle-VC configuration mode to configure VC specific parameters. • The VC is created upon exiting the mode. • **vpi**—ATM network virtual path identifier (VPI) for this PVC. The absence of the slash mark (/) and a VPI value defaults the value to 0. The **vpi** and **vci** keywords cannot both be set to 0. • **vci**—ATM network virtual channel identifier (VCI) for this PVC. The value range is from 0 to 1 less than the maximum value set for this interface by the **atm vc-per-vp** command. **Note** The **pvc-bundle** command is used instead of the **pvc** command to avoid the ambiguity between this command and the interface **pvc** command.
Step 4	**class** *class-name* **Example:** `Router(config-if-atm-member)# class control-class`	Attaches a VC class to this VC.
Step 5	**ubr** *pcr* **Example:** `Router(config-if-atm-member)# ubr 10000`	Configures UBR QoS and specifies the output PCR for the VC bundle member.
Step 6	**ubr+** *pcr mcr* **Example:** `Router(config-if-atm-member)# ubr+ 10000 3000`	Configures UBR+ QoS with a PCR and an MCR for the VC bundle member.

	Command or Action	Purpose
Step 7	**vbr-nrt** *pcr scr* [*mbs*] **Example:** `Router(config-if-atm-member)# vbr-nrt 20000 10000 32`	Configures VBR-nrt QoS with a PCR, an SCR, and MBS.
Step 8	**mpls experimental** [**other** | *range*] **Example:** `Router(config-if-atm-member)# mpls experimental 7`	Defines the experimental levels for packets to be forwarded on this PVC. • **other**—Any precedence level from 0 to 7 not explicitly configured. • *range*—An MPLS EXP level specified as a number or a range of numbers. Ranges can be expressed with a hyphen (2-5, for example), and numbers and ranges can be expressed in groups separated by commas; 1, 3, 5-7, for example.
Step 9	**bump** {**implicit** | **explicit** *precedence-level* | **traffic**} **Example:** `Router(config-if-atm-member)# bump explicit 7`	Specifies the bumping rule for the VC. • **implicit**—Applies the implicit bumping rule, which is the default, to a single VC or PVC bundle member or to all VCs in the bundle (VC class mode). The implicit bumping rule stipulates that bumped traffic is to be carried by a VC or PVC with a lower precedence level. • **explicit** *precedence-level*—Specifies a precedence level from 0 to 7 for the traffic to be bumped to. • **traffic**—Specifies that the VC or PVC accepts bumped traffic (the default condition). The **no** form of this command stipulates that the VC or PVC does not accept any bumped traffic.
Step 10	**protect** {**group** | **vc**} **Example:** `Router(config-if-atm-member)# protect vc`	Configures a VC class with protected group or protected VC status for application to a VC bundle member. • This command makes a bundle member part of the protected group of a bundle or a protected VC in a bundle. • **group**—Configures the VC or PVC bundle member as part of the protected group of the bundle. • **vc**—Configures the VC or PVC member as individually protected.
Step 11	**exit** **Example:** `Router(config-if-atm-member)# exit`	Exits the current configuration mode. • Continue entering **exit** at the prompt to exit each configuration mode.

Verifying ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection Configuration

Use the commands in the following steps as needed, to verify configurations for the MPLS EXP Bits Based ATM PVC Bundles VC Selection feature.

SUMMARY STEPS

1. **enable**
2. **debug atm bundle adjacency**
3. **debug atm bundle error**
4. **debug atm bundle events**
5. **debug atm bundle inarp**
6. **show atm bundle** [*bundle-name*]
7. **show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug atm bundle adjacency** **Example:** `Router# debug atm bundle adjacency`	Displays information about adjacency events such as addition, removal, and update of adjacencies for the bundle.
Step 3	**debug atm bundle error** **Example:** `Router# debug atm bundle error`	Displays debug messages for PVC bundle errors.
Step 4	**debug atm bundle events** **Example:** `Router# debug atm bundle event`	Displays bundle events such as when VC bumping occurs, when the bundle goes up or down, and so on.
Step 5	**debug atm bundle inarp** **Example:** `Router# debug atm bundle inarp`	Displays information about Inverse ARP events and errors on the bundle.

	Command or Action	Purpose
Step 6	**show atm bundle** [*bundle-name*] **Example:** `Router# show atm bundle new-york`	Displays the bundle attributes assigned to each VC member and the current working status of the VC members.
Step 7	**show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table detail`	Displays the contents of the MPLS FIB.

Configuration Examples for ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection

This section contains the following examples:

Configuring MPLS: Example

The following example shows how to configure MPLS:

```
ip cef
mpls ldp advertise labels
!
interface atm 0/0/3
 ip address 10.13.11.3 255.255.0.0
 mpls ip
 bundle bundle1
```

Defining ATM VC Classes and Parameters: Example

In the following example, VC classes are defined with parameters applicable to individual VCs in the bundle. Each VC class is preceded by a description of how it will be used.

```
! The following commands define the bundle class. Any bundle that uses this class will
! have AAL5snap encapsulation, broadcast on, use of Inverse ARP to resolve IP addresses,
! and OAM enabled at the bundle class level in the inheritance chain.
! This router uses IS-IS as an IP routing protocol.
!
router isis
 net 49.0000.0000.0000.1111.00
!
vc-class atm bundle-class
 encapsulation aal5snap
```

```
 broadcast
 protocol ip inarp
 oam-bundle manage 3
 oam 4 3 10
!
! The following VC class defines the parameters applicable to an individual VC
! in a bundle. The control-class carries precedence 7 traffic and it takes the
! bundle down when it is down. It uses the implicit bumping rule.
! The QoS is set to VBR-nrt.
!
vc-class atm control-class
 mpls experimental 7
 protect vc
 vbr-nrt 10000 5000 32
!
! The following VC class defines a premium class that carries precedence level 6 and 5
! traffic. It does not allow other traffic to be bumped onto it. The VC will choose
! precedence 7 VC as the alternate VC for its traffic when it goes down, and it belongs
! to the protected group of the bundle. The QoS type is VBR-nrt.
!
vc-class atm premium-class
 mpls experimental 6-5
 no bump traffic
 protect group
 bump explicit 7
 vbr-nrt 20000 10000 32
!
! The following VC class defines a priority class that carries precedence levels
! 4 through 2 traffic, uses the implicit bumping rule, allows bumped traffic,
! and belongs to the protected group of the bundle. The QoS type is UBR+.
!
vc-class atm priority-class
 mpls experimental 4-2
 protect group
 ubr+ 10000 3000

! The following VC class defines a basic-class that carries the traffic of the precedence
! levels not specified in the profile; it is part of a protected group.
! The QoS type is UBR.
!
vc-class atm basic-class
 mpls experimental other
 protect group
 ubr 10000
```

Associating an ATM VC Bundle with the Interface: Example

The following interface has one bundle, new-york, for connecting to three neighbors: new-york, san-francisco, and los-angeles. The new-york and san-francisco bundles have four members and los-angeles has three members.

```
interface atm 1/0.1 multipoint
 ip address 10.0.0.1 255.255.255.0
 ip router isis
 bundle new-york
!
! The following commands enable IP and OSI traffic flows in the bundle. The protocol ip
! command takes precedence over the protocol ip inarp command in the bundle class,
! according to the inheritance rule. The protocol clns command is configured so IP routing
! can be integrated. The OSI routing packets will go on the highest precedence VC in the
! bundle, while the OSI data packets, if any, will use the lowest precedence VC in the
```

```
! bundle. Other protocols such as IPX or AppleTalk, if configured, would always use the
! lowest precedence VC in the bundle.
  protocol ip 10.10.1.2 broadcast
  protocol clns 49.0000.0000.0000.2222.00 broadcast
  class bundle-class
!
! The following commands show how to configure the PVC bundles, including adding a VC
! to a bundle as a member.
  pvc-bundle ny-control 207
   class control-class
  pvc-bundle ny-premium 206
   class premium-class
  pvc-bundle ny-priority 204
   class priority-class
  pvc-bundle ny-basic 201
   class basic-class
 bundle san-francisco
  protocol clns 49.0000.0000.0000.3333.00 broadcast
  inarp 1
  class bundle-class
  pvc-bundle sf-control 307
   class control-class
  pvc-bundle sf-premium 306
   class premium-class
  pvc-bundle sf-priority 304
   class priority-class
  pvc-bundle sf-basic 301
   class basic-class
 bundle los-angeles
  protocol ip 1.1.1.4 broadcast
  protocol clns 49.0000.0000.0000.4444.00 broadcast
  inarp 1
  class bundle-class
  pvc-bundle la-high 407
   precedence 7-5
   protect vc
   class premium-class
  pvc-bundle la-mid 404
   precedence 4-2
   protect group
   class priority-class
  pvc-bundle la-low 401
   precedence other
   protect group
   class basic-class
!
The following commands configure PVC la-other as a standalone VC that does not belong to
any of the bundles.
!
 pvc la-other 400
  no protocol ip inarp
  broadcast
```

Creating a VC Class: Example

In the following example, a class called class1 is created and then applied to the bundle called bundle1:

```
! The following commands create the class class1:
vc-class atm class1
 encapsulation aal5snap
 broadcast
 protocol ip inarp
```

```
 oam-bundle manage 3
 oam 4 3 10
!
! The following commands apply class1 to the bundle called bundle1:
bundle bundle1
 class-bundle class1
```

With hierarchy precedence rules taken into account, VCs belonging to the bundle named bundle1 will be characterized by these parameters: AAL5SNAP (RFC1483 bridging) encapsulation, broadcast on, use of Inverse ARP to resolve IP addresses, and OAM enabled.

Additional References

The following sections provide references related to the MPLS EXP Bits Based ATM PVC Bundles VC Selection feature.

Related Documents

Related Topic	Document Title
QoS	• *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3 • *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3
MPLS	• *Cisco IOS Switching Services Configuration Guide*, Release 12.3 • *Cisco IOS Switching Services Command Reference*, Release 12.3
ATM VCs	• *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.3 • *Cisco IOS Wide-Area Networking Command Reference*, Release 12.3
SVC bundles	*IP to ATM SVC Bundles for Class of Service (CoS) Mapping*, Cisco IOS Release 12.2(4)T feature module
MPLS LDP	*MPLS Label Distribution Protocol*, Cisco IOS Release 12.2(4)T feature module
ATM VC bundle management on Cisco 12000 Series Internet Routers	*ATM VC Bundle Management on Cisco 12000 Series 8-Port OC-3 STM-1 ATM Line Cards*, Cisco IOS Release 12.0(23)S feature module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents the following commands, which were modified for the Cisco IOS Release 12.0(29)S of the ATM PVC Bundle Enhancement — MPLS EXP-Based PVC Selection feature:

- **mpls experimental**
- **show mpls forwarding-table**

mpls experimental

To configure Multiprotocol Label Switching (MPLS) experimental (EXP) levels for a virtual circuit (VC) class that can be assigned to a VC bundle and thus applied to all VC members of that bundle, use the **mpls experimental** command in VC-class configuration mode. To remove the MPLS EXP levels from the VC class, use the **no** form of this command.

To configure the MPLS EXP levels for a VC member of a bundle, use the **mpls experimental** command in bundle-VC configuration mode. To remove the MPLS EXP levels from the VC, use the **no** form of this command.

mpls experimental [**other** | *range*]

no mpls experimental

Syntax Description

other	(Optional) Specifies any MPLS EXP levels in the range from 0 to 7 that are not explicitly configured. This is the default.
range	(Optional) A single MPLS EXP level specified as a number, or a range of levels, specified as a hyphenated range.

Defaults

Defaults to the **other** keyword, that is, any MPLS EXP levels in the range from 0 to 7 that are not explicitly configured.

Command Modes

VC-class configuration (for a VC class)
Bundle-VC configuration (for ATM VC bundle members)

Command History

Release	Modification
12.2(8)T	This command was introduced.
12.0(29)S	This command was integrated into Cisco IOS Release 12.0(29)S.

Usage Guidelines

Assignment of MPLS EXP levels to VC bundle members allows you to create differentiated service because you can distribute the MPLS EXP levels over the different VC bundle members. You can map a single level or a range of levels to each discrete VC in the bundle, thereby enabling VCs in the bundle to carry packets marked with different levels. Alternatively, you can configure a VC with the **mpls experimental other** command to indicate that it can carry traffic marked with levels not specifically configured for it. Only one VC in the bundle can be configured with the **mpls experimental other** command to carry all levels not specified. This VC is considered the default one.

To use this command in VC-class configuration mode, enter the **vc-class atm** global configuration command before you enter this command. This command has no effect if the VC class that contains the command is attached to a standalone VC, that is, if the VC is not a bundle member.

To use this command to configure an individual bundle member in bundle-VC configuration mode, first enter the **bundle** command to enact bundle configuration mode for the bundle to which you want to add or modify the VC member to be configured. Then use the **pvc-bundle** command to specify the VC to be created or modified and enter bundle-VC configuration mode.

VCs in a VC bundle are subject to the following configuration inheritance guidelines (listed in order of next highest MPLS EXP level):

- VC configuration in bundle-VC mode
- Bundle configuration in bundle mode (with the effect of assigned VC class configuration)
- Subinterface configuration in subinterface mode

Note If you are using an ATM interface, you must configure all MPLS EXP levels (ranging from 0 to 7) for the bundle. For this configuration, Cisco recommends configuring one member of the bundle with the **mpls experimental other** command. The **other** keyword defaults to any MPLS EXP level in a range from 0 to 7 that is not explicitly configured.

Examples

The following example configures a class named control-class that includes an **mpls experimental** command that, when applied to a bundle, configures all VC members of that bundle to carry MPLS EXP level 7 traffic. Note that VC members of that bundle can be individually configured with the **mpls experimental** command at the bundle-VC level, and would supervene.

```
vc-class atm control-class
 mpls experimental 7
```

The following example configures permanent virtual circuit (PVC) 401 (named control-class) to carry traffic with MPLS EXP levels in the range of 4 to 2, overriding the level mapping set for the VC through VC class configuration:

```
pvc-bundle control-class 401
 mpls experimental 4-2
```

Related Commands

Command	Description
bump	Configures the bumping rules for a VC class that can be assigned to a VC bundle.
bundle	Creates a bundle or modifies an existing bundle, and enters bundle configuration mode.
class-vc	Assigns a VC class to an ATM PVC, SVC, or VC bundle member.
protect	Configures a VC class with protected group or protected VC status for application to a VC bundle member.
pvc-bundle	Adds a VC to a bundle as a member and enters bundle-VC configuration mode to configure that VC bundle member.
ubr	Configures UBR QoS and specifies the output PCR for an ATM PVC, SVC, VC class, or VC bundle member.
vbr-nrt	Configures the VBR-nrt QoS and specifies the output PCR, output sustainable cell rate, and output maximum burst cell size for an ATM PVC, SVC, VC class, or VC bundle member.
vc-class atm	Creates a VC class for an ATM PVC, SVC, or ATM interface and enter VC-class configuration mode.

show mpls forwarding-table

To display the contents of the Multiprotocol Label Switching (MPLS) Forwarding Information Base (FIB), use the **show mpls forwarding-table** command in privileged EXEC mode.

show mpls forwarding-table [{*network* {*mask* | *length*} | **labels** *label* [**-** *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]

Syntax Description

network	(Optional) Destination network number.
mask	(Optional) IP address of the destination mask whose entry is to be shown.
length	(Optional) Number of bits in mask of destination.
labels *label - label*	(Optional) Displays only entries with the specified local labels.
interface *interface*	(Optional) Displays only entries with the specified outgoing interface.
next-hop *address*	(Optional) Displays only entries with the specified neighbor as the next hop.
lsp-tunnel *tunnel-id*	(Optional) Displays only entries with the specified label switched path (LSP) tunnel, or with all LSP tunnel entries.
detail	(Optional) Displays information in long form (includes length of encapsulation, length of MAC string, maximum transmission unit (MTU), and all labels).

Command Modes

EXEC

Command History

Release	Modification
11.1 CT	This command was introduced.
12.1(3)T	This command was modified to reflect new MPLS Internet Engineering Task Force (IETF) terminology and command-line interface (CLI) command syntax.
12.2(8)T	The command was modified to accommodate use of the MPLS experimental (EXP) level as a selection criterion for packet forwarding. The output display was modified to include a bundle adjacency field and EXP (vcd) values when the optional **detail** keyword is specified.
12.0(29)S	This command was integrated into Cisco IOS Release 12.0(29)S.

Usage Guidelines

The optional parameters described allow specification of a subset of the entire FIB.

Examples

The following is sample output from the **show mpls forwarding-table** command:

```
Router# show mpls forwarding-table

Local Outgoing      Prefix           Bytes tag Outgoing       Next Hop
tag   tag or VC     or Tunnel Id     switched  interface
26    Untagged      10.253.0.0/16    0         Et4/0/0        172.27.32.4
28    1/33          10.15.0.0/16     0         AT0/0.1        point2point
```

```
29    Pop tag       10.91.0.0/16      0         Hs5/0         point2point
      1/36          10.91.0.0/16      0         AT0/0.1       point2point
30    32            10.250.0.97/32    0         Et4/0/2       10.92.0.7
      32            10.250.0.97/32    0         Hs5/0         point2point
34    26            10.77.0.0/24      0         Et4/0/2       10.92.0.7
      26            10.77.0.0/24      0         Hs5/0         point2point
35    Untagged  [T] 10.100.100.101/32 0         Tu301         point2point
36    Pop tag       172.1.0.0/16      0         Hs5/0         point2point
      1/37          172.1.0.0/16      0         AT0/0.1       point2point

[T]     Forwarding through a TSP tunnel.
        View additional tagging info with the 'detail' option
```

The following is sample output from the **show mpls forwarding-table** command when you specify **the detail** keyword. If the MPLS EXP level is used as a selection criterion for packet forwarding, a Bundle adjacency exp (vcd) field is included in the display. This field includes the EXP value and the corresponding virtual circuit descriptor (vcd) in parentheses.

```
Router# show mpls forwarding-table detail

Local Outgoing      Prefix            Bytes tag Outgoing       Next Hop
tag   tag or VC     or Tunnel Id      switched  interface
16    Pop tag       1.0.0.6/32        0         AT1/0.1       point2point
      Bundle adjacency exp(vcd)
      0(1) 1(1) 2(1) 3(1) 4(1) 5(1) 6(1) 7(1)
      MAC/Encaps=12/12, MTU=4474, Tag Stack{}
      00010000AAAA030000008847
      No output feature configured
17    18            1.0.0.9/32        0         AT1/0.1       point2point
      Bundle adjacency exp(vcd)
      0(1) 1(1) 2(1) 3(1) 4(1) 5(1) 6(1) 7(1)
      MAC/Encaps=12/16, MTU=4470, Tag Stack{18}
      00010000AAAA030000008847 00012000
      No output feature configured
18    19            1.0.0.10/32        0        AT1/0.1       point2point
      Bundle adjacency exp(vcd)
      0(1) 1(1) 2(1) 3(1) 4(1) 5(1) 6(1) 7(1)
      MAC/Encaps=12/16, MTU=4470, Tag Stack{19}
      00010000AAAA030000008847 00013000
      No output feature configured
19    17            10.0.0.0/8         0        AT1/0.1       point2point
      Bundle adjacency exp(vcd)
      0(1) 1(1) 2(1) 3(1) 4(1) 5(1) 6(1) 7(1)
      MAC/Encaps=12/16, MTU=4470, Tag Stack{17}
      00010000AAAA030000008847 00011000
      No output feature configured
20    20            10.10.0.0/8         0        AT1/0.1       point2point
      Bundle adjacency exp(vcd)
      0(1) 1(1) 2(1) 3(1) 4(1) 5(1) 6(1) 7(1)
      MAC/Encaps=12/16, MTU=4470, Tag Stack{20}
      00010000AAAA030000008847 00014000
      No output feature configured
21    Pop tag       10.10.0.0/24        0        AT1/0.1       point2point
      Bundle adjacency exp(vcd)
      0(1) 1(1) 2(1) 3(1) 4(1) 5(1) 6(1) 7(1)
      MAC/Encaps=12/12, MTU=4474, Tag Stack{}
      00010000AAAA030000008847
      No output feature configured
22    Pop tag       1.0.0.4/32          0        Et2/3         40.0.0.4
      MAC/Encaps=14/14, MTU=1504, Tag Stack{}
      000427AD10430005DDFE043B8847
      No output feature configured
```

```
show mpls forwarding-table detail

Local     Outgoing      Prefix           Bytes tag      Outgoing     Next Hop
tag       tag or VC     or Tunnel Id     switched       interface
16        Untagged      20.0.0.0/8         0            AT1/0.2      point2point
          Bundle adjacency exp(vcd)
          0(4) 1(4) 2(4) 3(4) 4(4) 5(4) 6(4) 7(4)
          MAC/Encaps=0/0, MTU=4474, Tag Stack {}
          No output feature configured
          per-packet load-sharing
```

Table 1 describes the significant fields shown in the output.

Table 1 show mpls forwarding-table Field Descriptions

Field	Description
Local tag	Label assigned by this router.
Outgoing tag or VC	Label assigned by the next hop or virtual path identifier (VPI)/virtual channel identifier (VCI) used to get to the next hop. The entries that you can specify in this column include the following: • [T]—Means forwarding through an LSP tunnel. • "Untagged"—Means that there is no label for the destination from the next hop or that label switching is not enabled on the outgoing interface. • "Pop tag"—Means that the next hop advertised an implicit NULL label for the destination and that this router popped the top label.
Prefix or Tunnel Id	Address or tunnel to which packets with this label are going.
Bytes tag switched	Number of bytes switched with this incoming label.
Outgoing interface	Interface through which packets with this label are sent.
Next Hop	IP address of the neighbor that assigned the outgoing label.
Bundle adjacency exp (vcd)	Bundle adjacency information. The list of numbers that follows includes the MPLS EXP value and the corresponding VCD.
MAC/Encaps	Length in bytes of the Layer 2 header and length in bytes of the packet encapsulation, including the Layer 2 header and label header.
MTU	Maximum transmission unit (MTU) of the labeled packet.
Tag Stack	All the outgoing labels. If the outgoing interface is transmission convergence (TC)-ATM, the VCD is also shown.
00010000AAAA030000008847 00013000	The actual encapsulation in hexadecimal form. A space separates Layer 2 and the label header.